(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 684 549 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
02.11.2000 Bulletin 2000/44

(51) Int. Cl.[7]: **G06F 7/00**

(21) Application number: **94830240.1**

(22) Date of filing: **23.05.1994**

(54) Method for parallel processing of fuzzy logic inference rules and corresponding circuit architecture with fuzzy inputs and outputs

Verfahren zur parallellen Bearbeitung von Fuzzy-Logik-Inferenzregeln und übereinstimmende Schaltkreisarchitektur mit Fuzzy-Eingabe und -Ausgabe

Méthode à traitement en parallèle des règles d'inférence de logique floue et architecture de circuit correspondant avec des entrées et sorties floues

(84) Designated Contracting States:
DE FR GB IT

(43) Date of publication of application:
29.11.1995 Bulletin 1995/48

(73) Proprietor:
CO.RI.M.ME.
CONSORZIO PER LA RICERCA SULLA MICROELETTRONICA NEL MEZZOGIORNO
95121 Catania (IT)

(72) Inventors:
- **Giacalone, Biagio**
  I-91100 Trapani (IT)
- **Catania, Vincenzo**
  I-95030 S. Agata Li Battiati (CT) (IT)
- **Luzzi, Claudio**
  I-87041 Acri (Cosenza) (IT)
- **Matranga, Vincenzo**
  I-90146 Palermo (IT)

(74) Representative: **Botti, Mario**
Botti & Ferrari S.r.l.
Via Locatelli, 5
20124 Milano (IT)

(56) References cited:
EP-A- 0 361 403
EP-A- 0 472 921
EP-A- 0 594 222

- **IEEE TRANSACTIONS ON FUZZY SYSTEMS, vol.2, no.2, May 1994, NEW YORK pages 93 - 106 CATANIA ET AL 'A VLSI Fuzzy Inference Processor Based on a Discrete Analog Approach'**
- **IEEE EXPERT, vol.1, no.3, 1986, NEW YORK US pages 55 - 62 TOGAI ET AL 'Expert System on a Chip etc...'**

EP 0 684 549 B1

Description

Field of Application

[0001] The present invention relates to a method for calculation in parallel of multiple fuzzy logic inference rules.

[0002] The present invention concerns also a circuit architecture for implementation of the above method having fuzzy inputs and outputs.

[0003] Specifically the present invention relates to a method for parallel processing of multiple fuzzy logic inference rules organised in fuzzy sets or logical functions of multiple fuzzy sets comprising membership functions defined in a so-called discourse universe and said rules being configured essentially as IF-THEN rules with at least one antecedent preposition and at least one consequent implication and each preposition comprising at least one term of comparison between the membership functions and a plurality of input data and the terms being separated by logical operators.

Known Art

[0004] Fuzzy logic has now been established as a technique capable of supplying solutions for broad classes of control problems for which conventional techniques, e.g. those based on Boolean logic, have proven unsuited for providing acceptable performance at acceptable cost.

[0005] Fuzzy logic supplies a method of modelling the 'inaccurate' modes of reasoning typical of the human mind and which play however an essential role just in the human ability to make decisions under conditions of uncertainty.

[0006] Fuzzy logic operates on a linguistic description of reality using a particular class of variables termed 'linguistic variables'. To each variable can be syntactically joined a set of values dependent upon it which can take on different meanings depending on the context in which they are employed. Said values are found starting from a primary term which represents the variable, from one of its contraries, and from a series of so-called modifiers of the primary term, e.g. as described in European patent application no. 92830095.3.

[0007] Each value assigned to a linguistic variable is represented furthermore by a so-called fuzzy set, i.e. a possibilistic distribution function which links each value of the variable in the corresponding definition domain known also as universe of discourse.

[0008] The functions which identify a fuzzy set in the universe of discourse of a variable are called membership function FA. The assembly of all the fuzzy sets of a linguistic variable is called 'term set'.

[0009] Membership functions are defined by means of a sample representation obtained by dividing the definition domain in m points and the interval [0, 1] in 1 levels.

[0010] At present, definition or memorisation in a fuzzy logic based electronic controller of the membership functions which identify the fuzzy sets represents one of the major restraints on development of new fuzzy logic applications and thus limits the theoretical potential of this methodology.

[0011] Indeed, if for implementation on hardware of the membership functions it is desired that said functions reflect the semantics of the fuzzy concept so as to obtain a correct incidence of a term in a rule, one is forced to use a considerable memory space. This makes fuzzy logic advantageous only for those applications where the term set of the linguistic variable consists of a reduced number of membership functions.

[0012] The data for a membership function are normally stored in a memory word. In known devices the memory area occupied is thus negatively influenced by the number of data necessary for defining these membership functions.

[0013] In many cases it has proven sufficient to memorise triangular membership functions generally not symmetrical or at most trapezoid membership functions so as to reduce the number of data necessary for their memorisation.

[0014] With these triangular or trapezoid membership functions type it is not at all necessary to memorize the values of the function at all points of the universe of discourse but it is sufficient to memorize only the points where the curve changes slope and the value of this slope.

[0015] Among the membership functions appropriate logical operations - termed 'inferential' - which allow description of the behaviour of a system with the change in input parameters are performable. These operations are performed by mean of fuzzy rules which have generally a syntax of the following type:

IF X IS A, THEN Y IS B

where X is the input value, A and B are membership functions FA which represent system knowledge, and Y is the output value.

[0016] The part of the rule preceding the term THEN is called the 'left' or 'antecedent' part while the following part is called 'right' or 'consequent' part of the inference rule.

[0017] The implication between the antecedent part and the consequent part of a fuzzy rule is governed by two laws:

- modus ponens: in it the truth of the implication (Th), i.e. of the consequent part of the rule, depends on that of the premise (Hp), i.e. the antecedent part of the rule;
- modus tollens: in it occurrence of the implication (Th) which ensures correctness of the premise (Hp).

[0018] Adopting the modus ponens as the rule, the degree of truth of the entire rule cannot be greater than that of the antecedent part.

[0019] Since the antecedent part can be made up of one or more terms T corresponding to hypotheses of the type (F is F') on the data F and on the membership functions F' its overall degree of truth which we shall indicate by the symbol Ω in the following description depends just on the inference operations on these same terms T.

[0020] In addition the overall degree of truth Ω takes on a determinate value by applying to these terms T the logical operators AND, OR and NOT.

[0021] The electronic data processing tools which allow performance of this type of operation must be provided with a particular architecture expressly dedicated to the set of inference operations which constitute the fuzzy logic computational model.

[0022] In a processor operating with fuzzy logic procedures there must be room for a circuit capable of calculating the overall degree of truth Ω regardless of the logical operators present.

[0023] Heretofore multivalue fuzzy logic inferences were calculated in different ways.

[0024] In the project developed at the OMRON by T. Yamakawa et al the inference processing circuit can operate analogically in parallel only on four rules whose antecedent part can have at most three terms.

[0025] In addition to this initial limitation, for design simplicity other constraints were imposed:

- the terms T of the antecedent part of the rules can be separated only by logical operators AND;
- the membership functions I' of the term sets of the input variables I can have only an S, Z, trapezoid or triangular shape;
- the inputs are deterministic, i.e. they correspond to an individual point P in the universe of discourse U.

[0026] The architecture of H. Watanabe et al performs in parallel all the rules for the same output variable. The user is however limited in his choice of the variables with which he can work. These can be only four input variables and two output variables out of fifty-one rules, or two input variables and one output out of one hundred two rules.

[0027] A plurality of Watanabe circuits can be connected in cascade under control of a software programme in such a manner as to process more than one hundred two rules. In this case moreover it is possible to introduce a feedback of the output signal on the input of one of the components.

[0028] In like manner circuits of this type can be connected to operate with a larger number of input variables.

[0029] These architectures however involve a dramatic increase in the silicon area occupied by the memories.

[0030] A third known solution consists of the Fuzzy Micro Controller of Neural Logix in which are used only symmetrical and linear membership functions (triangles, trapezes, etc.). Since each antecedent part of a rule can contain up to a maximum of sixteen terms, there are sixteen fuzzifiers at the input of this circuit.

[0031] With reference to triangular or trapezoid membership functions FA, by weight α_I of a set of data I for an antecedent part term T represented in the universe of discourse U by means of a membership function I' is meant the greatest value of the intersection between the input data set I and the membership function I' corresponding to said term T.

[0032] The Neural Logix circuit can process up to sixty-four rules. At input there can be applied the variables to be controlled or the fedback output variables.

[0033] In this processing circuit a neuronal network performs the smallest of the sixteen terms contained in the antecedent part of the rule. Among the overall degrees of truth Ω of all sixty-four antecedent parts is then calculated the maximum value by means of a circuit consisting of a single register which is continually updated on the basis of each evaluation of the weight of each antecedent part.

[0034] Lastly, a processor known in trade as 'WARP' and manufactured by the same applicant processes sequentially up to two hundred fifty-six rules whose antecedent parts are made up of four terms.

[0035] The architecture of the inferential part was designed to calculate the degree of truth of the premise by means of parallel computation on four α values. These weights α are taken simultaneously from the data memory once the input variables are known.

[0036] In the case of rules whose antecedent parts contain more than four terms T separated by logical operators the processing is carried out by dividing said antecedent parts in several antecedent sub-parts each of which contains four terms in the antecedent part allowing for the partial truth level w of each antecedent sub-part obtained by means of a feedback to the inference calculation circuit.

[0037] All the circuits heretofore available to the technicians of the industry cannot be considered absolutely effective because their efficacy depends heavily on the type of application.

[0038] In particular, the architectures which give priority to parallel processing of the inference rules in such a manner as to gain processing time lose necessarily in occupied silicon area.

[0039] On the other hand reduction of the occupied memory area by a decrease in the number of computational units causes efficiency of parallel processing to depend heavily on the number of rules associated with each individual inference operation.

[0040] Actually, if all the inferences to be processed are characterised by the same number NFR of fuzzy

rules there can be a less than optimal use of available resources each time the number of processing units NPU present in the architecture is not exactly a submultiple of the number NFR of fuzzy rules.

[0041] In this case the following relationship is not verified:

N_{FR} mod $N_{PU} = 0$

i.e. N_{FR} is not exactly divisible by the number NPU.

[0042] In practice it is not always possible to introduce a number of inferential units equal to the number of the rules describing the process. Typically one is forced to oversize or undersize the calculation structure.

[0043] Known from the article to Togai et al, published on IEEE EXPERT, vol. 1, No. 3, 1986 and entitled: "Expert System on a Chip: an Engine for Real-Time Approximate Reasoning" is a full parallel processor according to the prior art knowledge. It is necessary a plurality of such integrated circuit in order to process a plurality of rules.

[0044] In particular this document is directed to a fuzzy logic inference engine that processes fuzzy logic rules having an antecedent with only one term. All the rules are accessed in parallel, and each individual rule is accessed in serial fashion.

[0045] Another article to Catania et al., published on IEEE TRANSACTION ON FUZZY SYSTEMS, vol. 2, No. 2, May 1994, and entitled: "A VLSI Fuzzy Inference Processor Based on a Discrete Analog Approach" describes a mixed (analog-digital) processor, that computes in parallel antecedent variables and serially rules.

[0046] Finally, the European Patent Application No. 0 594 222 relates to an analog fuzzy architecture that processes both rules and variables in parallel. In this way a plurality of computational units are needed to process a plurality of rules.

[0047] The technical problem underlying the present invention is to identify a new parallel processing method for multiple fuzzy rules which would not depend on the number of terms making up the antecedent part of the rules or the logical operators linking them.

Summary of the invention

[0048] The solution ideal underlying the present invention is to provide simultaneous processing of several rules which could be configured dynamically in a flexible manner on the basis of the characteristics of the different applications for which the circuit might be designed.

[0049] On the basis of this solution idea the technical problem is solved by means of a parallel processing method as defined in the characterising part of claims 1 and following.

[0050] The technical problem is solved also by equipment described in claims 7 and following.

[0051] The characteristics and advantages of the method in accordance with the present invention are set forth in the description of an embodiment thereof given below by way of non-limiting example with reference to equipment illustrated in the annexed drawings.

Brief Description of the Drawings

[0052] In the drawings:

FIGS. 1A-1D show membership functions I' of a possible term set and a set of input data I,

FIG. 2 shows schematically a circuit architecture provided in accordance with the present invention for implementation of the multiple fuzzy rule parallel processing method,

FIG. 3 shows in greater detail the circuitry architecture of FIG. 2, and

FIG. 4 shows a variant embodiment of a detail of the architecture of FIG. 3.

Detailed Description

[0053] With reference to figures 1A-1D a membership function I' of a linguistic or logical variable M is represented by means of a vectorial system where along the abscissa axis is defined a so-called universe of discourse U while along the ordinate axis is defined a so-called degree of truth or membership G.

[0054] The input data I are represented by the same reference system.

[0055] In FIGS. 1A-1D are shown four membership functions I' which identify in the universe of discourse U as many fuzzy sets which are part of a so-called term set. In FIGS. 1A-1D are indicated also the weights α_I of each term T of an inference rule R, i.e. the highest value of the intersection between the set of input data I and the membership function I' corresponding to said term T.

[0056] In practice the input variables I can be reduced to a single value P in the universe of discourse U (then termed 'crisp' input). In these cases the truth level αj of each term Tj is nothing but the value of the membership function I' corresponding to that predetermined input value P.

[0057] The method in accordance with the present invention calls for the use of a new inferential unit 1 which allows inferential processing of Nr fuzzy logic inference rules each of which can be made up of Na antecedent terms with a single consequent term.

[0058] This unit 1 operates on fuzzy inputs and for each point of the universe of discourse U receives the greatest value of the intersection between the input data I and the corresponding membership functions I'.

[0059] By way of example let us consider a rule R1 formed as follows:

IF (A is A') AND (NOT B is B') OR NOT [(C is C') AND (D is D')] THEN

The antecedent part of this rule R1 is made up of four terms Tj placed between round parentheses and takes on a value determined by applying thereto the logical operators present AND, OR and NOT.

[0060] In the graphs of FIGS. 1A-1D are shown the weights α_A, α_B, α_C and α_D for the four terms Tj :

$$\alpha_A = \text{maxx}\,[\,\text{min}\,(A, A')]$$

$$\alpha_B = \text{maxx}\,[\,\text{min}\,(\text{not } B, B')]$$

$$\alpha_C = \text{maxx}\,[\,\text{min}\,(C, C')]$$

$$\alpha_D = \text{maxx}\,[\,\text{min}\,(D, D')]$$

[0061] In fuzzy logic semantics, with the operators AND and OR are associated respectively minimum and maximum operations between two or more elements while with the operator NOT is associated a complementary operation for one in the universe U.

[0062] The operations for the two calculation stages in which the method in accordance with the present invention is divided for processing of a fuzzy logic inference rule R are the following:

- the weight α of each term T of the antecedent part of the fuzzy logic inference rule R is estimated,
- the overall degree of truth Ω of the rule R is updated.

[0063] After calculating the overall degree of truth Ω for the antecedent part of the fuzzy rule the method proceeds with computation of the consequent part of said rule.

[0064] The method in accordance with the present invention provides for simultaneous (pipeline) processing of these calculation stages. In the time employed for calculation of the overall degree of truth Ω of a rule R of a given sequence of rules there is determined the weight α' of the terms T' following the antecedent part of the rule R.

[0065] When all the rules for the same inference have been processed the method provides for composition of a fuzzy logic output Y by supplying a level of truth for each point of the universe of discourse U.

[0066] The computation unit 1 is thus compatible with itself since it operates on fuzzy inputs and produces outputs of the same type which can be used directly as inputs of another inferential unit of the same type or defuzzified.

[0067] FIG. 2 shows schematically an example of circuitry architecture designed for parallel processing of multiple fuzzy logic inference rules.

[0068] The circuitry architecture of FIG. 2 comprises an inferential unit 1 with modular structure and comprising a plurality of same lines 2 for inferential processing connected in parallel in a number equal to the fuzzy logic inference rules Ri to be processed simultaneously.

[0069] Each inferential processing line 2 provides computation of a fuzzy logic inference rule Ri and comprises a first evaluation block 3, a second calculator block 4 and a third processing block 5 for the membership functions of the consequent part based on the value of the output of the calculator block 4.

[0070] The evaluation block 3 has a plurality of inputs 6 and one output 7. Said inputs 6 receive through a plurality of busses 8 the values of the terms T of the antecedent part of the fuzzy logic inference rule R.

[0071] The output 7 is connected through the bus 9 to an input 10 of the calculator block 4 and supplies the weight α of the terms T of the antecedent part of the fuzzy logic inference rule R.

[0072] The calculator block 4 has an output 11 connected through the bus 12 to a first input 13 of the processing block 5. The output 11 supplies the overall truth level Ω of the antecedent part of the fuzzy logic inference rule R.

[0073] Said processing block 5 has an input 14 which receives through the bus 15 the membership functions Xi which characterise the consequent part of the fuzzy logic inference rule R.

[0074] Furthermore the processing block 5 has an output 16 connected through bus 17 to an input 18 of another block 19 for union. The outputs 16 of the processing blocks 5 supply the value of the membership functions of the consequent part processed starting from the value coming from the calculator block 4.

[0075] The union block 19 has an output 20 which supplies through bus 21 the fuzzy logic value Y of the group of Nr fuzzy logic inference rule Ri for a description variable of the physical universe which is the object of the study developed by the inferential unit 1.

[0076] FIG. 3 shows in greater detail a first embodiment of a circuit of the diagram proposed in FIG. 2.

[0077] In the proposed embodiment reference is made to the defuzzification method denominated 'Max-Min' which operates by cutting off the membership function. The circuitry architecture in accordance with the present invention can still be applied in the presence of other defuzzification methods of known type.

[0078] In the example of FIG. 3 the data which were present on the busses 8 of FIG. 2 are conveyed by means of a single bus 22 on which travel the system input variables A, B, C....

[0079] The evaluation block 3 comprises a first minimiser block 23 which has two inputs 24 and 25 and one output 26.

[0080] The input 25 of the minimiser block 23 receives through bus 22 the input data A while the input 24 receives through bus 27 the membership functions Ai for the linguistic variable A and in like manner for the linguistic variables B, C, etc.

[0081] The output 26 is connected through the bus 28 to an input 29 of a sliding register 30. The sliding reg-

ister 30 receives on a second input 31 a clock signal CK and has an input 32 connected through the bus 33 to a first input 34 of another maximiser block 35.

[0082] The maximiser block 35 has 3 inputs 34, 36 and 37 and one output 38. The second input 36 of the maximiser block 35 receives the clock signal CK while the third input 37 is connected through the bus 41 to an output 39 of another sliding register 40.

[0083] The sliding register 40 has two inputs 42 and 43. The first input 42 receives the clock signal CK while the second input 43 is connected through the bus 44 to the output 38 of the maximiser block 35.

[0084] The output 38 of the maximiser block 35 is connected to a first input 45 of a first sliding register 46 included in the calculator block 4.

[0085] The sliding register 46 receives on another input 47 the clock signal CK and has an output 48 connected through the bus 49 to a first input 51 of a maximinimiser block 50.

[0086] The maximinimiser block 50 receives on a second input 52 the clock signal CK and has a third input 53 connected through the bus 54 to an output 56 of another sliding register 55 and an output 57 connected through the bus 58 to a first input 59 of the sliding register 55.

[0087] A second input 60 of the sliding register 55 receives the clock signal CK.

[0088] The output 57 of the maximinimiser block 50 is connected to an input 61 of another sliding register 62 which receives at a second input 63 the clock signal CK and is connected in cascade through the bus 64 to a first input 66 of a minimiser block 65.

[0089] The minimiser block 65 receives on a second input 67 through the bus 15 the membership functions Xi which characterise the consequent part of the fuzzy logic inference rule R.

[0090] The minimiser block 65 is connected in cascade through the bus 17 to the union block 19.

[0091] We shall now discuss operation of this circuitry architecture.

[0092] The bus 22 carries the system input variables (A, B, C...). For each point where the universe of discourse is divided said variables are compared in parallel with the respective values of the membership functions for linguistic variables A1, A2, ..., AN.

[0093] At the end of this first processing phase which is performed a number of times equal to the number of points of discretisation of the universe of discourse U on the bus 44 at the output of the first evaluation block 3 there is the weight SIMBOLO 97 \f "Symbol"§Ai of the first term Ai of the antecedent part of the fuzzy logic inference rule Ri.

[0094] One then proceeds to evaluation of the weight α_B for the second antecedent term B of all the rules related to the membership functions B1, B2, ..., BN.

[0095] This new value is supplied to the second calculator stage 4 which compares it with the weight α_{Ai} for the previous term of the antecedent part on the basis of the logical operation AND or OR present in the fuzzy logic inference rule Ri.

[0096] Based on this comparison there is calculated the partial truth level Ω and the overall truth level Ω is updated.

[0097] The system proceeds in accordance with this sequence until all the terms Ti present in the antecedent part of the fuzzy logic inference rule Ri have been processed.

[0098] At the end of the process, on each line 12 are present the overall truth levels Ω of every fuzzy logic inference rule R processed while on each inferential processing line 2 is present the processed value of the membership functions of the consequent part.

[0099] In accordance with the method in accordance with the present invention this circuitry architecture allows parallel processing of multiple fuzzy rules with a high level of configurability.

[0100] Indeed, reconfiguration mechanisms capable of adapting the system to the characteristics of the application can be introduced. Indeed, the number Nr of inferential processing lines 2 depends essentially on the processing in question and hence, in accordance with the user's choices, it is possible to configure the structure in the desired manner.

[0101] It is also possible to group the Nr simultaneously processable rules from each individual computation unit depending on the exigencies of the moment.

[0102] The further structure configuration flexibility can be achieved by sending on the bus 15 the data Xi for the membership functions of the terms of the consequent part of appropriately selected sets of rules.

[0103] The circuitry diagram of FIG. 3 operates on rules in which a level of priority among logical operators has not been instituted.

[0104] It is possible to allow for a priority among these logical operators by means of a circuit in which the sliding register 55 in the calculator block 4 is doubled to allow separately for the values deriving from OR or AND logical operations.

[0105] FIG. 4 shows an alternative embodiment of the calculator block 4 in which it is possible to define a priority among the logical operators.

[0106] In the circuit of FIG. 4 the sliding register 55 is split into an OR register 55A and an AND register 55B.

[0107] These registers OR 55A and AND 55B receive at an input, 68 and 69 respectively, the clock signal CK.

[0108] A second input, 70 and 71 respectively, is connected through the busses 72 and 73 to respective outputs 74 and 75 of a path selector 80.

[0109] The outputs 76 and 77 of the registers OR 55A and AND 55B are connected through busses 78 and 79 respectively to inputs 81 and 82 of the maximinimiser block 50.

[0110] The path selector 80 has two inputs 83 and

84 connected through the busses 85 and 86 respectively to two outputs 87 and 88 respectively of the maximinimiser block 50.

[0111] The output 88 of the maximinimiser block 50 is connected to a first input 89 of a sliding register 90 which receives on a second input 91 the clock signal CK.

[0112] The sliding register 90 has an output 92 connected through the bus 58 to the input of processing block 5 for the membership functions of the consequent part.

[0113] By means of the circuitry structure of FIG. 4 it is possible to calculate the partial truth levels w for the logical operation with higher priority level to compare them only later in accordance with the logical operation with lower priority level.

[0114] The path selector 80 permits separation in the corresponding registers of the partial truth levels w for OR or AND logical operators.

[0115] Lastly it is possible to insert in input and output blocks which allow the structure to operate on deterministic magnitudes.

[0116] The input circuit is reduced to a simple block which, given the deterministic input, produces a fuzzy output with a single, substantially crisp, value.

[0117] In like manner the output circuit is a defuzzifier circuit which transforms a fuzzy value into a deterministic output. This transformation can be adapted to the accuracy required for the type of application.

[0118] The proposed circuitry diagram is compatible with itself, i.e. it exhibits inputs and outputs of the fuzzy type or, by introduction of appropriate input and output blocks, of the deterministic type.

Claims

1. Method of parallel processing of multiple fuzzy logic inference rules (R) organised in fuzzy sets or logical functions of multiple fuzzy sets that include corresponding membership functions (I') defined in a universe of discourse (U) and said inference rules (R) being configured as IF-THEN rules with at least one antecedent part and at least one consequent part and each part comprising at least one term (T) of comparison between membership functions (I') and a plurality of input data (I) and the terms (T) being separated by logical operators (OL), and characterised in that the method comprises the steps of:

   A - In parallel for each fuzzy logic inference rule, serially evaluating the terms (T) of, the antecedent of that rule, by means of the steps of:

   A1 - calculating a weight (α) of a term (T) of the antecedent part as the greatest value of the intersection between the set of input data (I) and the corresponding membership functions (I'); and

   A2 - generating a partial truth level according to said calculated weight of the term (T) of said antecedent part and the previous partial truth level, wherein the last partial truth level in the generated series is the overall truth level for that fuzzy logic inference rule (R), and

   B - providing a fuzzy logic signal based on the overall truth level for each of the fuzzy logic inference rules.

2. Processing method in accordance with claim 1 characterised in that the step of calculating includes the step of calculating the intersection at all points (m) of the universe of discourse (U).

3. Processing method in accordance with claim 1 characterised in that the step of generating comprises the step of updating a register with the partial truth level.

4. Processing method in accordance with claim 3 characterised in that the step of calculating and the step of generating occur simultaneously for at least two fuzzy logic inference rules.

5. Processing method in accordance with claim 1 characterised in that the step of generating includes the step of presenting at least one of the overall truth level as a fuzzy value.

6. Processing method in accordance with claim 1 characterised in that the logical operators seperating the terms (T), are OR, AND and NOT and are associated with a maximum, minimum and complementary operation, respectively.

7. Circuitry architecture for parallel processing of multiple fuzzy logic inference rules (R) organised in fuzzy sets or logical functions of multiple fuzzy sets that include corresponding membership functions (I') defined in a universe of discourse (U) and said inference rules (R) being configured as IF-THEN rules with at least one antecedent part and at least one consequent part and each part comprising at least one term (T) of comparison between membership functions (I') and a plurality of input data (I) and the terms (T) being separated by logical operators (OL), the circuitry comprising:

   - at least one inferential unit (1),
   - at least one data bus (22), and

- at least one union block (19);

characterised in that:

A - said inferential unit (1) has a modular structure and comprises a plurality of identical inferential processing lines (2) for processing a respective plurality of fuzzy logic inference rules, connected in parallel between the data bus (22) and the union block (19), and each individual inferential processing line (2) comprises

A1 -- an evaluation block (3) for calculating a weight (α) of a term (T) of the antecedent part as the greatest value of the intersection between the set of input data (I) and the corresponding membership functions (I'); and A2 -- a calculator block (4) for generating a partial truth level according to said calculated weight (α) of the term (T) of said antecedent part and the previous partial truth level, so that the last in the generated series of said partial truth levels is the overall truth level for that inference rule (R), and

B - the union block (19) provides a fuzzy logic output based on said overall truth level for each fuzzy logic inference rule.

8. Circuitry architecture in accordance with claim 7 characterised in that each of the inferential processing lines (2) further comprises a block for processing of the membership functions of the consequent part (5).

9. Circuitry architecture in accordance with claim 8 characterised in that said blocks (3,4,5) are connected together in series.

10. Circuitry architecture in accordance with claim 9 characterised in that every processing block for the membership functions of the consequent part (5) of every inferential processing line (2) receives the series of partial truth levels of the fuzzy logic inference rule (R) and membership functions (X), the membership functions defining the consequent part of the fuzzy logic inference rule (R), and supplies an activation level (Ω_R) of the processed fuzzy logic inference rule (R).

11. Circuitry architecture in accordance with claim 8 characterised in that every evaluation block (3) comprises at least one first minimum calculator block (23), at least one first shift register (30), at least one second maximum calculator block (35) and at least one second shift register (40).

12. Circuitry architecture in accordance with claim 11 characterised in that said at least one minimum calculator block (23), at least one first shift register (30) and at least one second maximum calculator block (35) are connected in series together and said at least one second shift register (40) has an output (39) connected to an input (37) of the maximum calculator block (35) and has an input (43) connected to an output (38) of the maximum calculator block (35).

13. Circuitry architecture in accordance with claim 8 characterised in that every calculator block (4) comprises at least one first shift register (46), at least one maximum-minimum calculator block (50) and at least one second shift register (55).

14. Circuitry architecture in accordance with claim 13 characterised in that said at least one first shift register (46) and at least one maximum-minimum calculator block (50) are connected in series together and said at least one second shift register (55) has an output (56) connected to an input (53) of the maximum-minimum calculator block (50) and has an input (59) connected to an output (57) of the maximum-minimum calculator block (50).

15. Circuitry architecture in accordance with claim 8 characterised in that every processing block for the membership functions of the consequent part (5) comprises at least one shift register (62) and at least one minimum calculator block (65).

16. Circuitry architecture in accordance with claim 15 characterised in that said at least one shift register (62) and at least one minimum calculator block (65) are connected in series together and said at least one minimum calculator block (65) receives the values of the membership functions (X) which characterise the consequent part of the fuzzy logic inference rule (R).

17. Circuitry architecture an accordance with claim 13 characterised in that it comprises two registers (55A,55B) which have an input (70,71) connected to a path selector (80) and an output (76,77) connected to an input (81,82) of the maximum-minimum calculator block (50).

18. Circuitry architecture in accordance with claim 17 characterised in that the path selector (80) has at least two inputs (83,84) connected to at least two outputs (87,88) of the maximum-minimum calculator block (50).

Patentansprüche

1. Verfahren zur Parallelverarbeitung mehrerer Fuzzy-

logik-Inferenzregeln (R), organisiert in Fuzzy-Mengen oder logischen Funktionen mehrerer Fuzzy-Mengen, die entsprechende Zugehörigkeitsfunktionen (I') enthalten, definiert in einem Diskursraum (U), wobei die Inferenzregeln (R) als WENN-DANN-Regeln mit mindestens einem Prämissenteil und mindestens einem Schlußteil ausgebildet sind und jeder Teil mindestens einen Term (T) des Vergleichs zwischen Zugehörigkeitsfunktionen (I') und mehreren Eingangsdaten (I) enthält, wobei die Terme (T) durch logische Operatoren (OL) getrennt sind, dadurch gekennzeichnet, daß das Verfahren folgende Schritte aufweist:

A - parallel für jede Fuzzy-Logik-Inferenzregel, serielles Auswerten der Terme (T) der Prämisse dieser Regel durch folgende Schritte:

A1 -- Berechnen eines Gewichts (() eines Terms (T) des Prämissenteils als größten Wert des Durchschnitts zwischen der Menge von Eingangsdaten (I) und den entsprechenden Zugehörigkeitsfunktionen (I'); und

A2 -- Erzeugen einer Teil-Wahrheitsebene entsprechend dem berechneten Gewicht des Terms (T) des Prämissenteils und der vorausgehenden Teil-Wahrheitsebene, wobei die letzte Teil-Wahrheitsebene in der erzeugten Serie die Gesamt-Wahrheitsebene für diese Fuzzy-Logik-Inferenzregel (R) ist,

B -- Bereitstellen eines Fuzzylogik-Signals basierend auf der Gesamt-Wahrheitsebene für jede der Fuzzy-Logik-Inferenzregeln.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Schritt des Berechnens den Schritt des Berechnens des Durchschnitts sämtlicher Punkte (m) des Diskursraums (U) beinhaltet.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Schritt des Erzeugens den Schritt des Aktualisierens eines Registers mit den Teil-Wahrheitsebenen beinhaltet.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß der Schritt des Berechnens und der Schritt des Erzeugens gleichzeitig für mindestens zwei Fuzzy-Logik-Inferenzregeln auftritt.

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Schritt des Erzeugens den Schritt des Darstellens mindestens einer der Gesamt-Wahrheitsebenen als Fuzzy-Wert beinhaltet.

6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die logischen Operatoren, die die Terme (T) trennen, (ODER, UND und NICHT und ARR) sind, in Verbindung mit einer Maximum-, Minimum- bzw. Komplementäroperation.

7. Schaltungsaufbau für die Parallelverarbeitung von mehreren Fuzzy-Logik-Inferenzregeln (R), organisiert in Fuzzy-Mengen oder logischen Funktionen mehrerer Fuzzy-Mengen, die entsprechende Zugehörigkeitsfunktionen (I') enthalten, definiert in einem Diskursraum (U), wobei die Inferenzregeln (R) als WENN-DANN-Regeln mit mindestens einem Prämissenteil und mindestens einem Schlußteil ausgebildet sind und jeder Teil mindestens einen Term (T) des Vergleichs zwischen Zugehörigkeitsfunktionen (I') und mehreren Eingangsdaten (I) enthält, wobei die Terme (T) durch logische Operatoren (OL) getrennt sind, wobei die Schaltung umfaßt:

- mindestens eine Inferenzeinheit (1),
- mindestens einen Datenbus (22), und
- mindestens einen Vereinigungsblock (19);

dadurch gekennzeichnet, daß:

A - die Inferenzeinheit (1) eine modulare Struktur aufweist und mehrere identische Inferenz-Verarbeitungsleitungen (2) aufweist, um eine zugehörige Mehrzahl von Fuzzy-Logik-Inferenzregeln zu verarbeiten, parallelgeschaltet zwischen dem Datenbus (22) und dem Vereinigungsblock (19), wobei jede einzelne Inferenzverarbeitungsleitung (2) aufweist:

A1 -- einen Auswerteblock (3) zum Berechnen eines Gewichts (() eines Terms (T) eines Prämissenteils als den größten Teil des Durchschnitts zwischen der Menge von Eingangsdaten (I) und den entsprechenden Zugehörigkeitsfunktionen (I'); und

A2 -- einen Berechnungsblock (4) zum Erzeugen einer Teil-Wahrheitsebene entsprechend dem berechneten Gewicht (() des Terms (T) des Prämissenteils und der vorausgehenden Teil-Wahrheitsebenen, so daß die letzte in der erzeugten Folge von Teil-Wahrheitsebene die Gesamt-Wahrheitsebene für diese Inferenzregel (R) ist, und

B -- der Vereinigungsblock (19) ein Fuzzy-Logik-Ausgangssignal basierend auf der Gesamt-Wahrheitsebene für jede Fuzzy-Logik-Inferenzregel liefen.

8. Schaltungsaufbau nach Anspruch 7, dadurch

gekennzeichnet, daß jede der Inferenz-Verarbeitungsleitungen (2) außerdem einen Block zum Verarbeiten der Zugehörigkeitsfunktionen des Schlußteils (5) aufweist.

9. Schaltungsaufbau nach Anspruch 8, dadurch gekennzeichnet, daß die Blöcke (3, 4, 5) in Serie geschaltet sind.

10. Schaltungsaufbau nach Anspruch 9, dadurch gekennzeichnet, daß jeder Verarbeitungsblock für die Zugehörigkeitsfunktionen des Schlußteils (5) jeder Inferenz-Verarbeitungsleitung (2) die Folge von Teil-Wahrheitsebenen der Fuzzy-Logik-Inferenzregel (R) und die Zugehörigkeitsfunktionen (X) empfängt, wobei die Zugehörigkeitsfunktionen den Schlußteil der Fuzzy-Logik-Inferenzregel (R) definieren, um einen Aktivierungspegel ((R) der verarbeiteten Fuzzy-Logik-Inferenzregel (R) zu liefern.

11. Schaltungsaufbau nach Anspruch 8, dadurch gekennzeichnet, daß jeder Auswertungsblock (3) mindestens einen Minimumberechnungsblock (23), mindestens ein erstes Schieberegister (30), mindestens einen zweiten Maximumberechnungsblock (35) und mindestens ein zweites Schieberegister (40) aufweist.

12. Schaltungsaufbau nach Anspruch 11, dadurch gekennzeichnet, daß der mindestens eine Minimumberechnungsblock (23), das mindestens eine erste Schieberegister (30) und der mindestens eine zweite Maximumberechnungsblock (35) in Serie geschaltet sind, und daß das mindestens eine zweite Schieberegister (40) mit seinem Ausgang (39) an den Eingang (37) des Maximumberechnungsblocks (35) angeschlossen ist, und mit seinem Eingang (43) an dem Ausgang (38) des Maximumberechnungsblocks (35) angeschlossen ist.

13. Schaltungsaufbau nach Anspruch 8, dadurch gekennzeichnet, daß jeder Berechnungsblock (4) mindestens ein erstes Schieberegister (46), mindestens einen Maximum-Minimum-Berechnungsblock (50) und mindestens ein zweites Schieberegister (55) enthält.

14. Schaltungsaufbau nach Anspruch 13, dadurch gekennzeichnet, daß das mindestens eine erste Schieberegister (46) und mindestens der eine Maximum-Minimum-Berechnungsblock (50) in Reihe geschaltet sind, und das mindestens eine zweite Schieberegister (55) mit einem Ausgang (56) an einen Eingang (53) des Maximum-Minimum-Berechnungsblocks (50) angeschlossen und mit einem Eingang (59) an einen Ausgang (57) des Maximum-Minimum-Berechnungsblocks (50) angeschlossen ist.

15. Schaltungsaufbau nach Anspruch 8, dadurch gekennzeichnet, daß jeder Verarbeitungsblock für die Zugehörigkeitsfunktionen des Schlußteils (5) mindestens ein Schieberegister (62) und mindestens einen Minimumberechnungsblock (65) aufweist.

16. Schaltungsaufbau nach Anspruch 15, dadurch gekennzeichnet, daß das mindestens eine Schieberegister (62) und der mindestens eine Minimumberechnungsblock (65) in Serie geschaltet sind und der mindestens eine Minimumberechnungsblock (65) die Werte der Zugehörigkeitsfunktionen (X) empfängt, welche den Schlußteil der Fuzzy-Logik-Inferenzregel (R) kennzeichnen.

17. Schaltungsaufbau nach Anspruch 13, dadurch gekennzeichnet, daß er zwei Register (55A, 55B) enthält, die mit einem Eingang (70, 71) an einen Wegselektor (80) und mit einem Ausgang (76, 77) an einen Eingang (81, 82) des Maximum-Minimum-Berechnungsblocks (50) angeschlossen sind.

18. Schaltungsaufbau nach Anspruch 17, dadurch gekennzeichnet, daß der Wegselektor (80) mindestens zwei Eingänge (83, 84) aufweist, die an mindestens zwei Ausgänge (87, 88) des Maximum-Minimum-Berechnungsblocks (50) angeschlossen sind.

Revendications

1. Procédé de traitement en parallèle de règles d'inférence multiples en logique floue (R) organisées en ensembles flous ou fonctions logiques d'ensembles flous multiples qui comprennent des fonctions d'appartenance correspondantes (I') définies dans un univers de discours (U), les règles d'inférence (R) ayant la forme de règles SI-ALORS (IF-THEN) avec au moins une partie d'antécédent et au moins une partie de conséquent, chaque partie comprenant au moins un terme (T) de comparaison entre des fonctions d'appartenance (I') et une pluralité de données d'entrée (I), les termes (T) étant séparés par des opérateurs logiques (OL), caractérisé en ce que le procédé comprend les étapes suivantes :

   A - en parallèle pour chaque règle d'inférence en logique floue, évaluer an série les termes (T) de l'antécédent de cette règle par les étapes suivantes :

      A1 - calculer le poids a d'un terme (T) de la partie d'antécédent en tant que la plus grande valeur de l'intersection entre l'ensemble de données d'entrée (I) et les

fonctions d'appartenance correspondantes (I'), et

A2 - produire un niveau de vérité partielle selon le poids calculé du terme (T) de la partie d'antécédent et du niveau de vérité partielle précédant, le dernier niveau de vérité partielle de la série produite étant le niveau de vérité d'ensemble pour cette règle d'inférence an logique floue (R) ; et

B - fournir un signal de logique floue sur la base du niveau de vérité d'ensemble de chacune des règles d'inférence an logique floue.

2. Procédé de traitement selon la revendication 1, caractérisé an ce que l'étape de calcul comprend l'étape consistant à calculer l'intersection pour tous les points (M) de l'univers du discours (U).

3. Procédé de traitement selon la revendication 1, caractérisé en ce que l'étape de production comprend l'étape consistant à mettre à jour un registre par le niveau de vérité partielle.

4. Procédé de traitement selon la revendication 3, caractérisé en ce que l'étape de calcul et l'étape de production surviennent simultanément pour au moins deux règles d'inférence en logique floue.

5. Procédé de traitement selon la revendication 1, caractérisé en ce que l'étape de production comprend l'étape consistant à présenter au moins un niveau de vérité d'ensemble comme valeur floue.

6. Procédé de traitement selon la revendication 1, caractérisé en ce que les opérateurs logiques séparant les termes (T) sont OU, ET et NON, et sont respectivement associés à une opération de maximum, de minimum et de complément.

7. Architecture de circuit logique pour traitement parallèle de règles d'inférence multiples en logique floue (R) organisées en ensembles flous ou fonctions logiques d'ensembles flous multiples qui comprennent des fonctions d'appartenance correspondantes (I') définies dans un univers de discours (U), les règles d'inférence (R) ayant la forme de règles SI-ALORS (IF-THEN) avec au moins une partie d'antécédent et au moins une partie de conséquent, chaque partie comprenant au moins un terme (T) de comparaison entre des fonctions d'appartenance (I') et une pluralité de données d'entrée (I), les termes (T) étant séparés par des opérateurs logiques (OL), le circuit comprenant :

- au moins un module d'inférence (1) ;
- au moins un bus de données (22), et
- au moins un bloc d'union (19) ;

caractérisé en ce que :

A - le module d' inférence (1) a une structure modulaire et comprend plusieurs lignes de traitement d'inférence identiques (2) pour traiter une pluralité respectives de règles d'inférence en logique floue connectées en parallèle entre le bus de données (22) et le bloc d'union (19), et chaque ligne de traitement d'inférence (2) comprend :

A1 - un bloc d'évaluation (3) pour calculer le poids (a) d'un terme (T) de la partie d'antécédent en tant que la plus grande valeur de l'intersection entre l'ensemble de données d'entrée (I) et les fonctions d'appartenance correspondantes (I'), et

A2 - un bloc de calcul (4) pour produire un niveau de vérité partielle selon le poids calculé (α) du terme (T) de la partie d'antécédent et le niveau de vérité partielle précédent, de sorte que le dernier de la série produite des niveaux de vérité partielle est le niveau de vérité d'ensemble pour cette règle d'inférence (R) ; et

B - le bloc d'union (19) fournit une sortie de logique floue sur la base du niveau de vérité d'ensemble pour chaque règle d'inférence de logique floue.

8. Architecture de circuit selon la revendication 7, caractérisée en ce que chacune des lignes de traitement d'inférence (2) comprend en outre un bloc pour traiter les fonctions d'appartenance de la partie de conséquent (5).

9. Architecture de circuit selon la revendication 8, caractérisée en ce que les blocs (3, 4, 5) sont connectés en série.

10. Architecture de circuit selon la revendication 9, caractérisée en ce que chaque bloc de traitement pour les fonctions d'appartenance de la partie de conséquent (5) de chaque ligne de traitement par inférence (2) reçoit la série de niveaux de vérité partielle de la règle d'inférence en logique floue (R) et des fonctions d'appartenance (X), les fonctions d'appartenance définissant la partie de conséquent de la règle d'inférence en logique floue (R), et fournit un niveau d' activation (Ω_R) de la règle d'inférence de logique floue traitée (R).

11. Architecture de circuit selon la revendication 8, caractérisée en ce que chaque bloc d'évaluation (3) comprend au ricins un premier bloc de calcul de minimum (23), au moins un premier registre à décalage (30), au moins un second bloc de calcul

de maximum (35) et au moins un second registre à décalage (40).

12. Architecture de circuit selon la revendication 11, caractérisée en ce que lesdits au moins un bloc de calcul de minimum (23), au moins un premier registre à décalage (30) et au moins un second bloc de calcul de maximum (35) sont connectés en série, et ledit au moins un second registre à décalage (40) a une sortie (39) connectée à une entrée (37) du bloc de calcul de maximum (35) et a une entrée (43) connectée à une sortie (38) du bloc de calcul de maximum (35).

13. Architecture de circuit selon la revendication 8, caractérisée en ce que chaque bloc de calcul (4) comprend au moins un premier registre à décalage (46), au moins un bloc de calcul de maximum-minimum (50), et au moins un second registre à décalage (55).

14. Architecture de circuit selon la revendication 13, caractérisée en ce que lesdits au moins un registre à décalage (46) et au moins un bloc de calcul de maximum-minimum (50) sont connectés en série, et ledit au moins un second registre à décalage (55) a une sortie (56) connectée à une entrée (53) du bloc de calcul de maximum-minimum (50) et a une entrée (59) connectée à une sortie (57) du bloc de calcul de maximum-minimum (50).

15. Architecture de circuit selon la revendication 9, caractérisée en ce que chaque bloc de traitement pair les fonctions d'appartenance de la partie de conséquent (5) comprend au moins un registre à décalage (62) et au moins un bloc de calcul de minimum (65).

16. Architecture de circuit selon la revendication 15, caractérisée en ce que lesdits au moins un registre à décalage (62) et au moins un bloc de calcul de minimum (65) sont connectés en série et ledit au moins un bloc de calcul de minimum (65) reçoit les valeurs des fonctions d'appartenance (X) qui caractérisent la partie de conséquent de la règle d'inférence de logique floue (R).

17. Architecture de circuit selon la revendication 13, caractérisée en ce qu'il comprend deux registres (55A, 55B) qui ont une entrée (70, 71) connectée à un sélecteur de trajet (80) et une sortie (76, 77) connectée à une entrée (81, 82) du bloc de calcul de maximum-minimum (50).

18. Architecture de circuit selon la revendication 17, caractérisée en ce que le sélecteur de trajet (80) a au moins deux entrées (83, 84) connectées à au moins deux sorties (87, 88) du bloc de calcul de maximum-minimum (50).

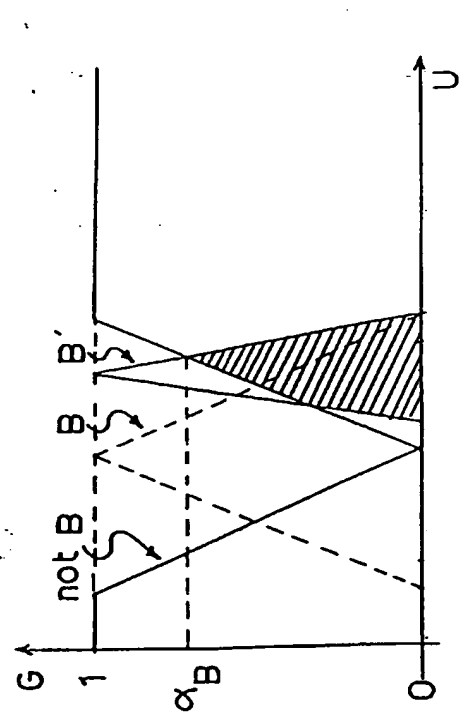

FIG. 1B

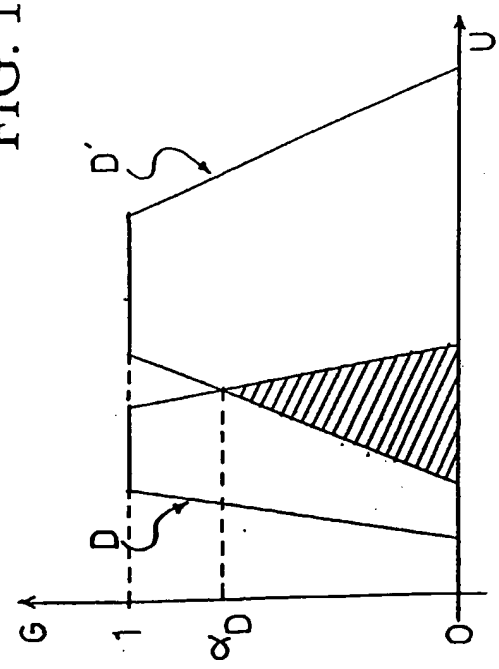

FIG. 1D

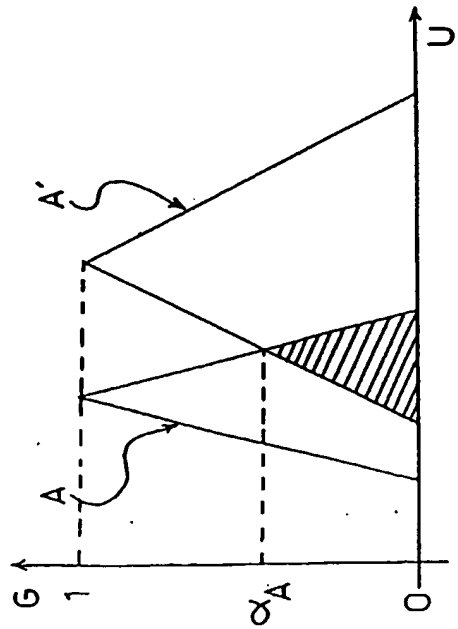

FIG. 1A

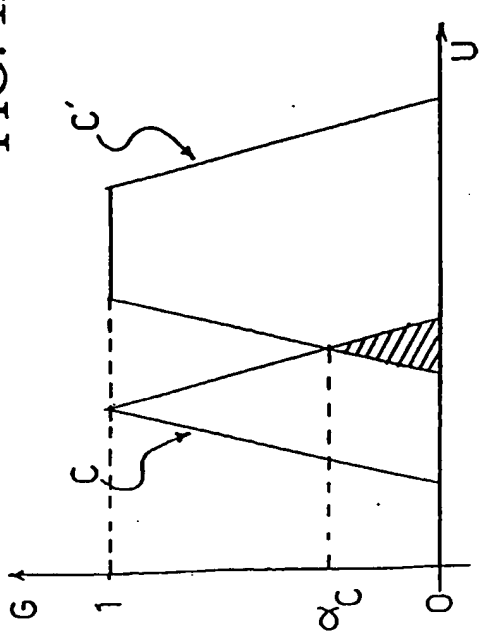

FIG. 1C

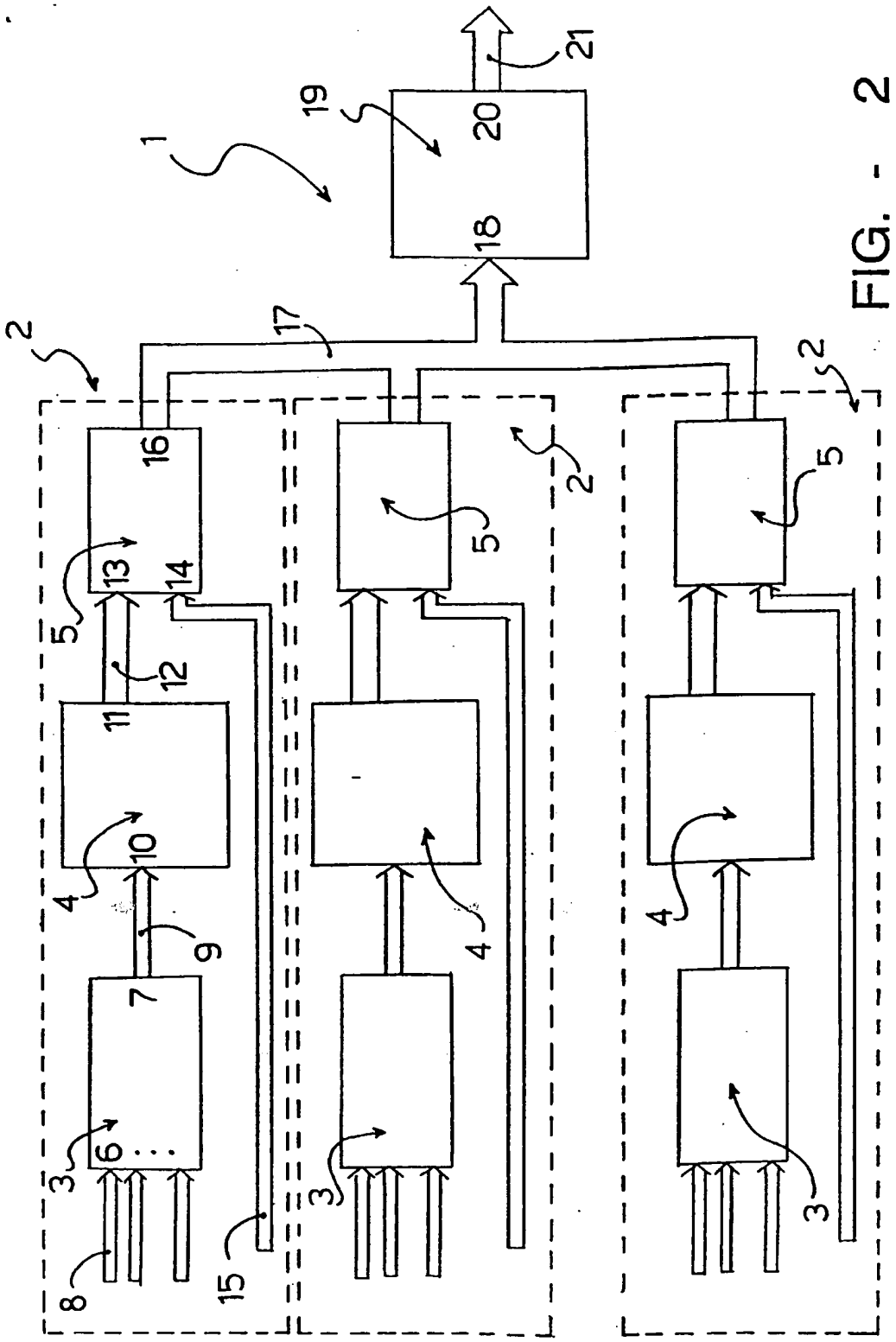

FIG. - 2

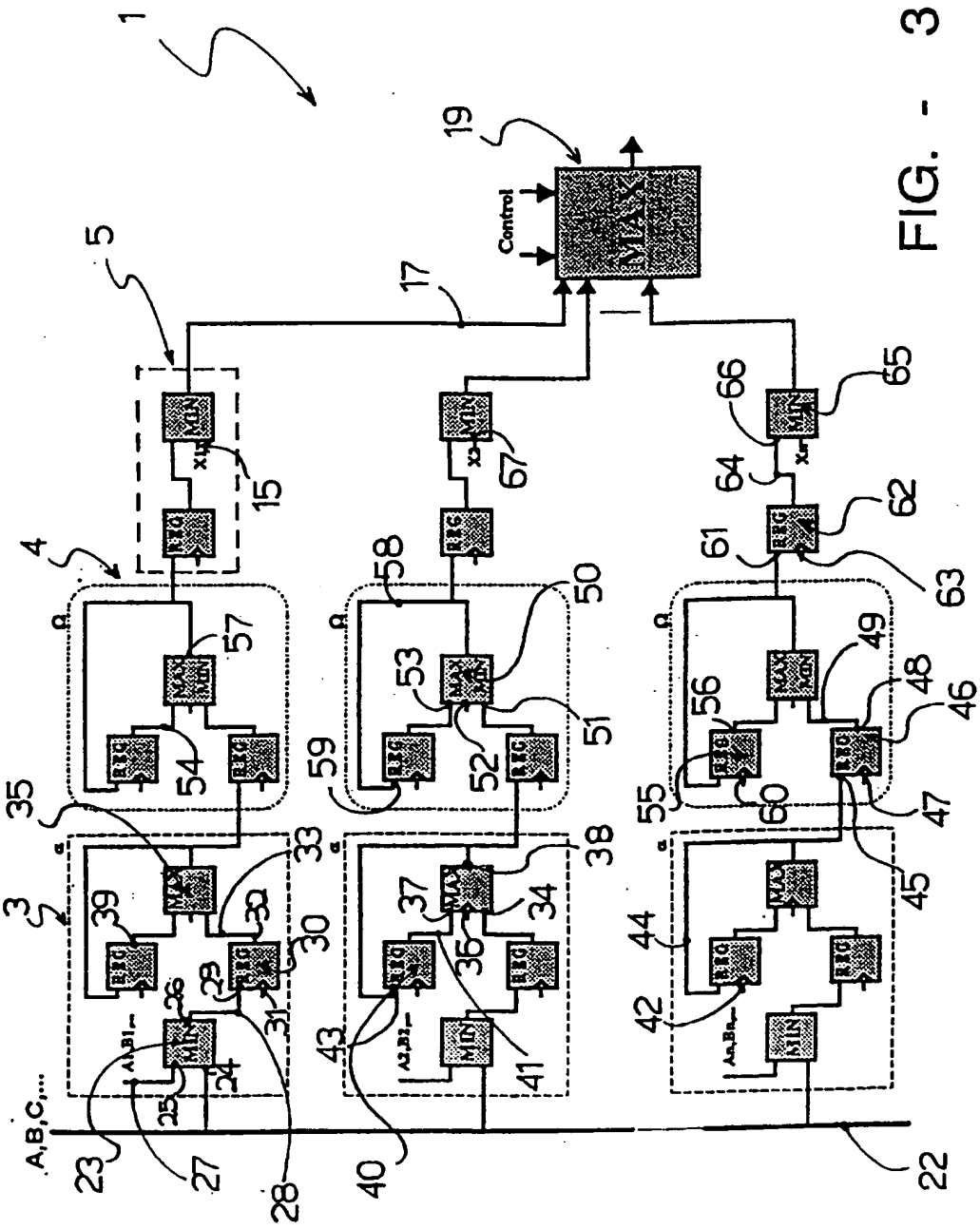
1
19
Control
MAX
5
17
4
15
67
65
66
64
62
61
63
58
50
57
53
49
56
48
46
59
51
52
54
60
55
47
35
33
38
3
39
37
34
45
44
30
36
29
42
43
41
24
25
31
A,B,C,....
23
27
28
40
22

FIG. - 3

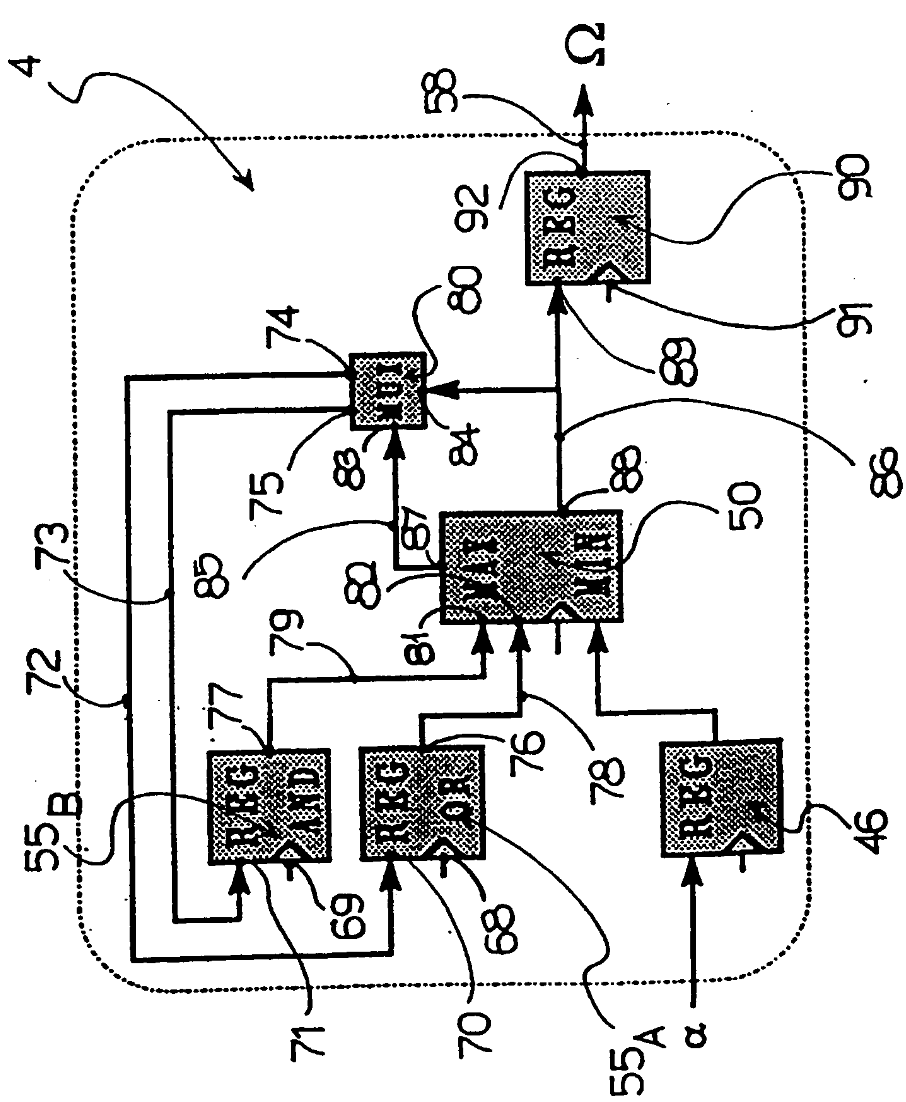
4
REG
AND
REG
OR
REG
REG
α
Ω
55B
55A
46
50
58
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
84
85
86
87
88
89
90
91
92

FIG. - 4